

# 1. Introduction

## 1.1 Purpose

This document outlines the deployment strategy for the "Time in My Pocket" application, ensuring a secure, scalable, and efficient transition from development to production.

## 1.2 Scope

The plan covers deployment environments, CI/CD pipelines, monitoring, security measures, backup strategies, and risk mitigation for the AI-driven time management application.

## 1.3 Target Audience

- DevOps Engineers
- System Administrators
- AI/ML Developers
- QA Engineers

# 2. Installation

**To download SQL Server Developer Edition, follow these steps:**

- Navigate to the official Microsoft SQL Server download page.
- Scroll down until you find the “Developer” edition section, then click the “Download now” button to begin the download.

**Or, download a free specialized edition**

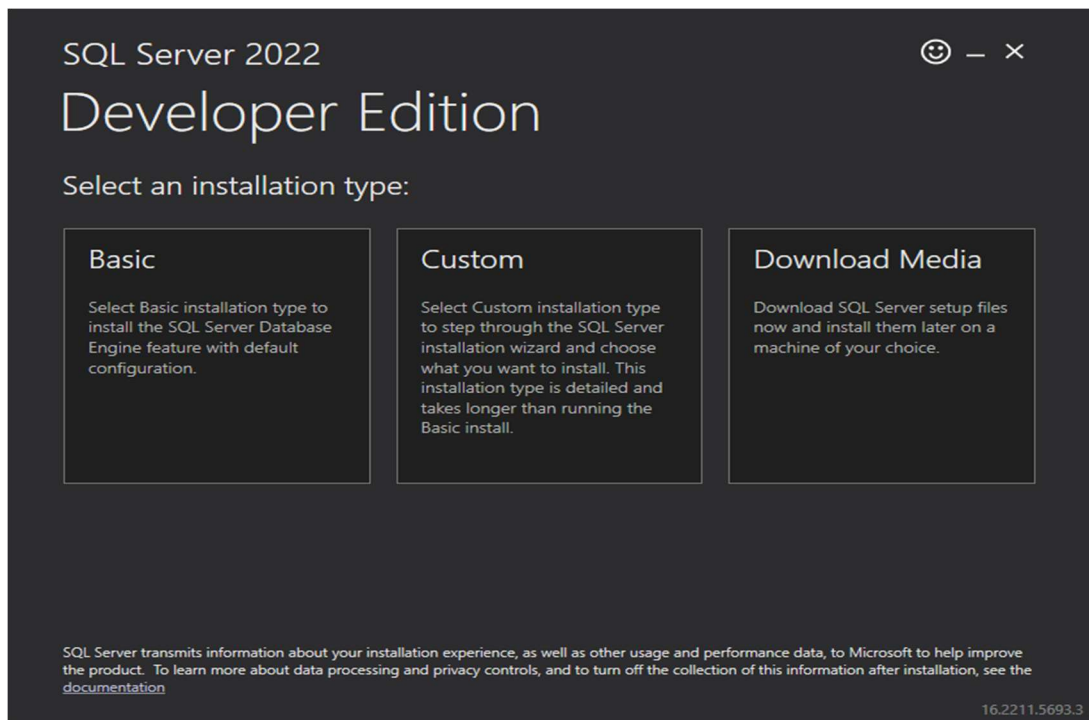
	
<b>Developer</b>	<b>Express</b>
<small>SQL Server 2022 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.</small>	<small>SQL Server 2022 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.</small>
<a href="#">Download now</a>	<a href="#">Download now</a>

**Launch the Installer:**

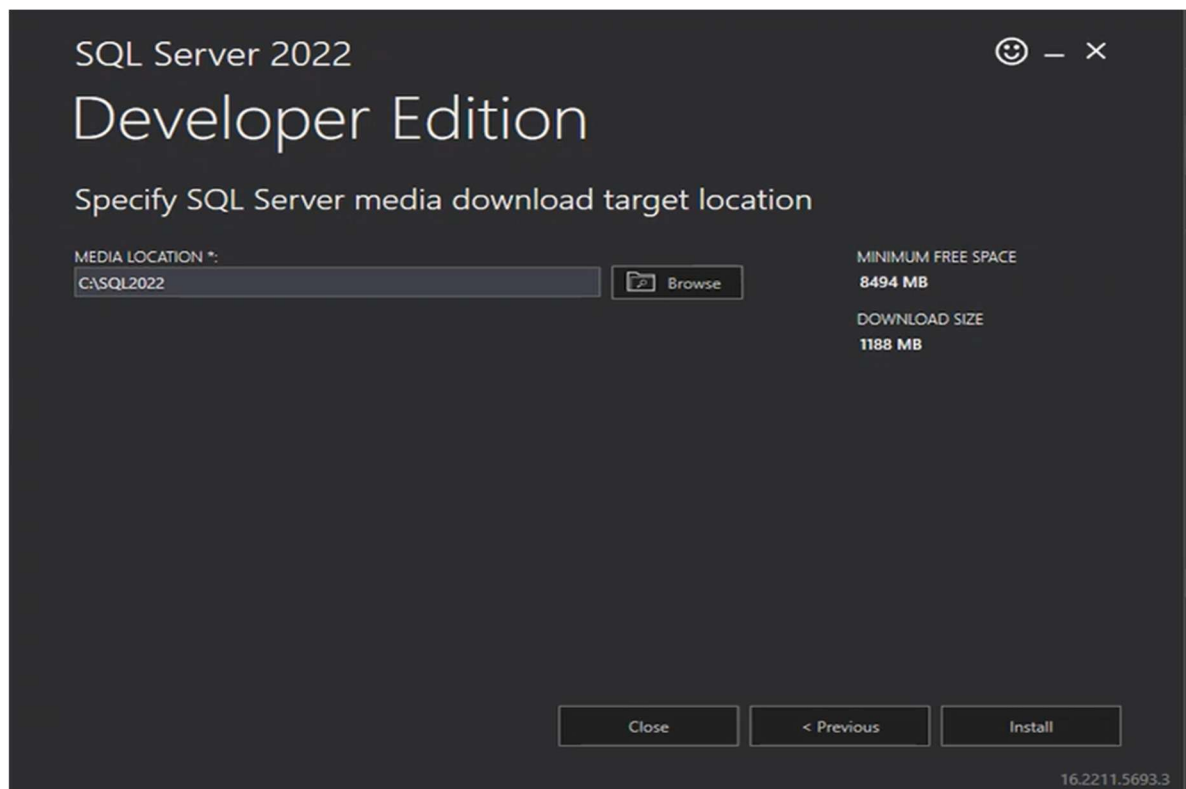
- Find the downloaded file named **SQL2022-SSEI-Dev.exe** on your computer.
- Double-click the file to initiate the installation process.

## Select the Installation Type:

- When the SQL Server Installation Center opens, choose the **"Custom"** option to proceed.

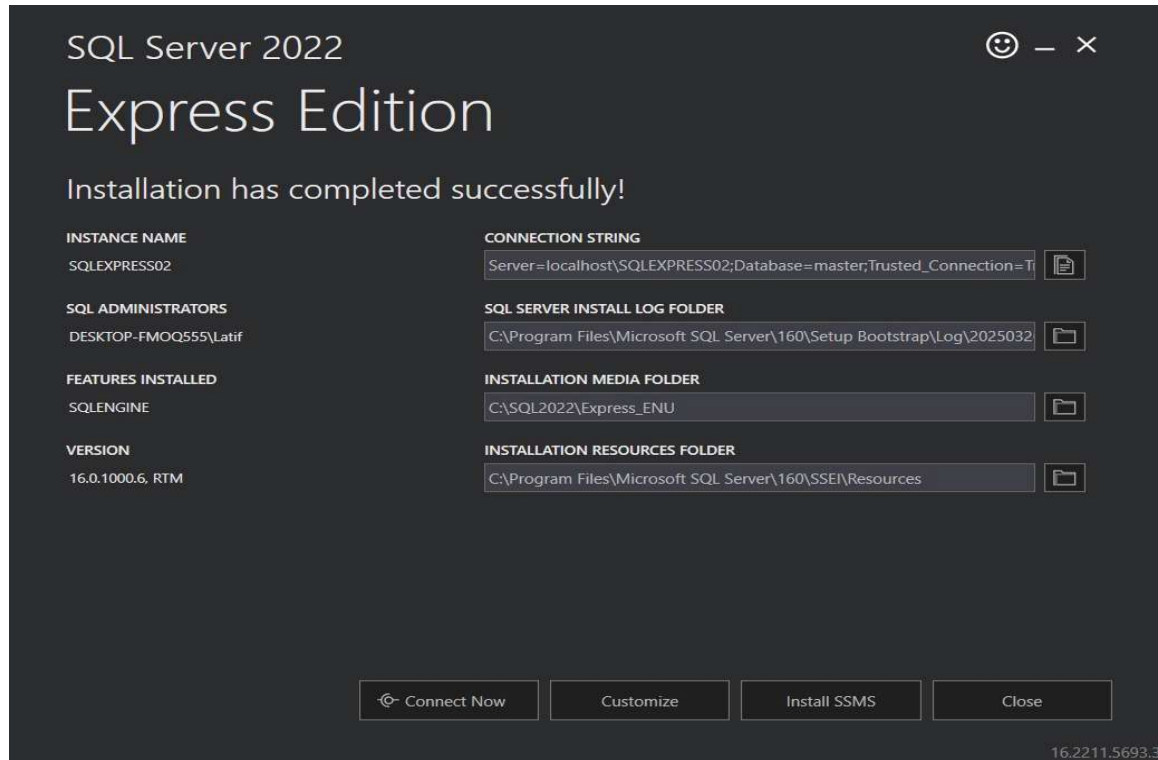


- Click on the "Install" button.



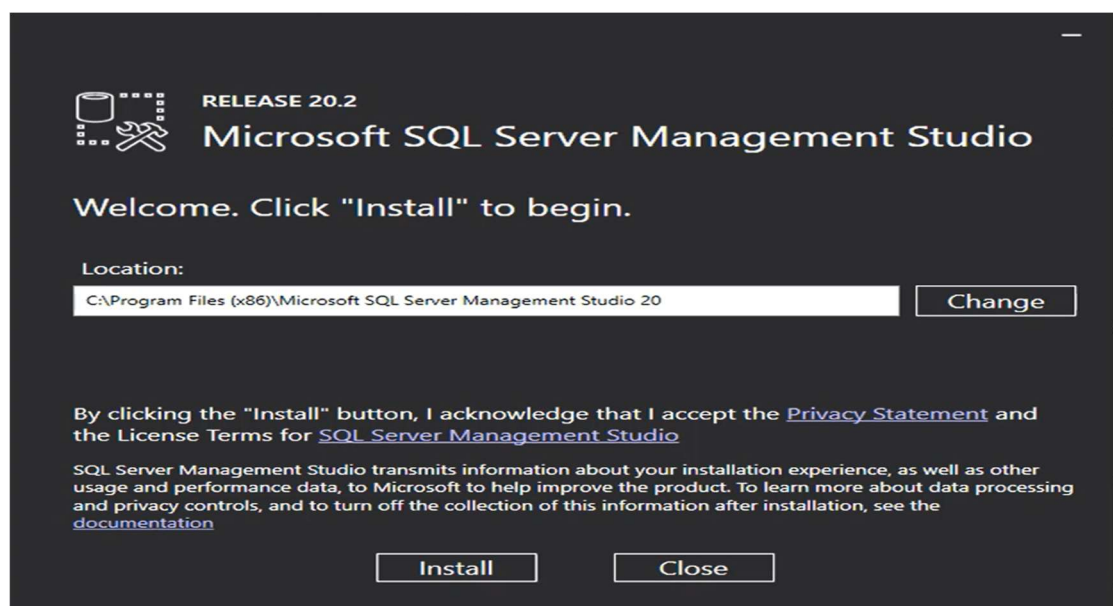
### Download SSMS:

- Click on the “Install SSMS button in the preceding step or visit the [Microsoft SSMS download](#) page.
- Click on the “Download SQL Server Management Studio(SSMS)” link.



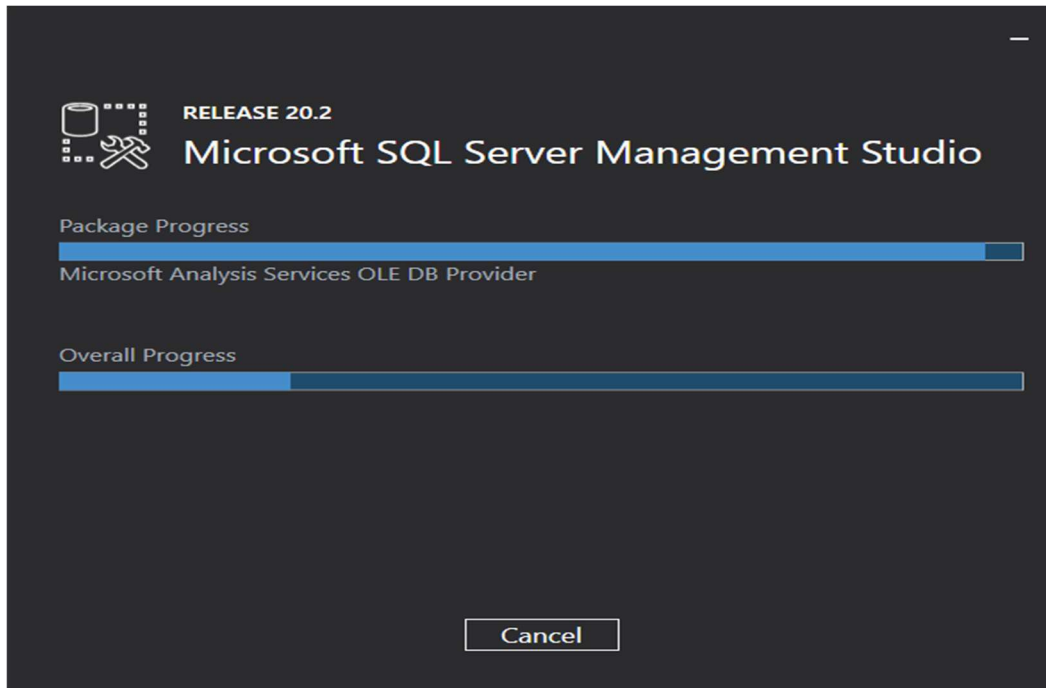
### Launch the SSMS Installer:

- Find the downloaded file named **SSMS-Setup-ENU.exe** on your computer.
- Double-click the file to open the installer, then click the "Install" button to begin the installation process.



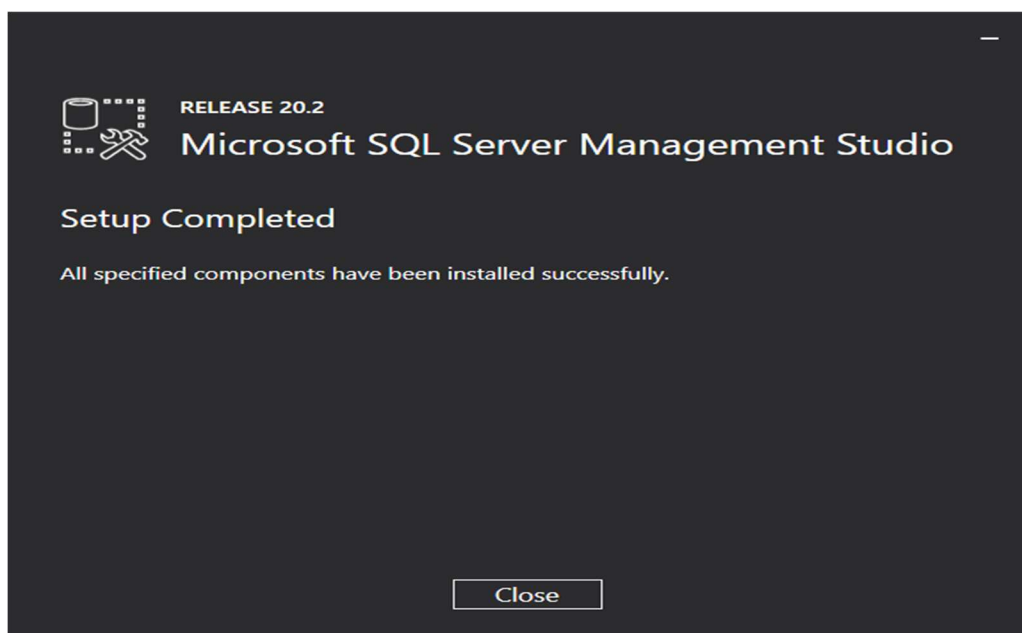
### Follow the Installation Wizard:

- Review and accept the license terms displayed in the setup wizard.
- Select the installation location (the default path is typically suitable).
- Click "**Install**" to start the installation process.



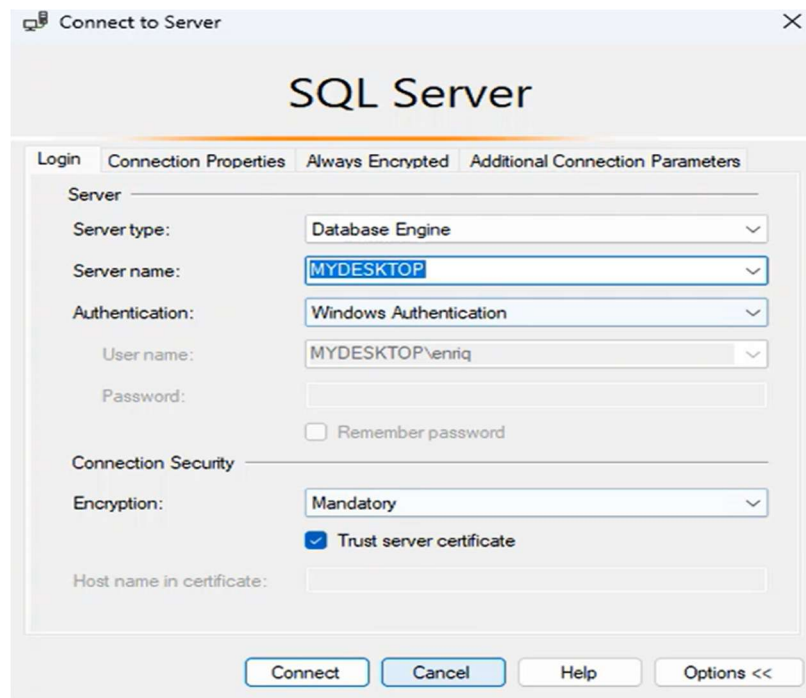
### Finish the Installation:

- After the installation is finished, click "**Close**" to exit the setup.
- You can now launch SQL Server Management Studio (SSMS) from the Start Menu.

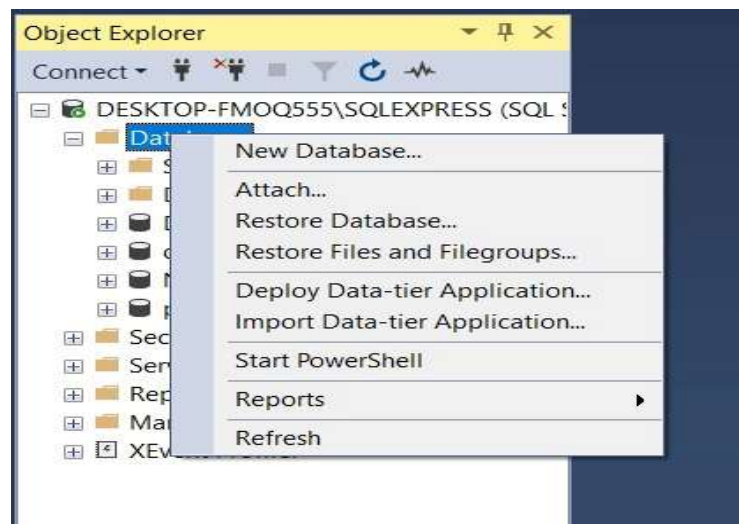


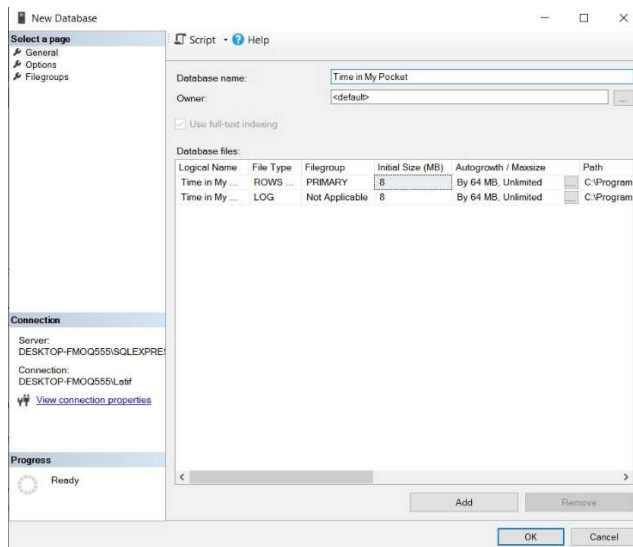
### Initial SSMS Setup:

- Click the **Start** button and find **SQL Server Management Studio** in the list of programs.
- When opening SSMS, select the option to log in using your **Windows credentials** to connect to SQL Server.



### Create a New Database in SSMS for MSSQL:





## 3. Deployment Architecture

### 3.1 Infrastructure Overview

- **Cloud Provider:** AWS (Primary) / Google Cloud (Backup)
- **Compute Services:** Kubernetes clusters with Docker containers for AI processing.
- **Database:** Microsoft SQL Server (Managed Instance) for user data and activity logs.
- **Storage:** S3 for static assets (articles, audio clips) and user-generated content.
- **Caching:** Redis for real-time suggestion caching.
- **Load Balancing:** AWS Elastic Load Balancer (ELB) with auto-scaling.
- **Monitoring & Logging:** Prometheus + Grafana for performance metrics; ELK Stack for logs.

### 3.2 Deployment Environments

Environment	Purpose	Hosting Platform
Development	Feature development and testing	AWS Dev Instance
Staging	Pre-production testing	AWS Staging Instance
Production	Live system for end users	AWS Production Instance

## 4. Deployment Process

### 4.1 Continuous Integration & Deployment (CI/CD)

**Tools:** GitHub Actions + Docker

**Steps:**

1. Code pushed to the main branch triggers automated tests (unit, integration, AI model validation).
2. Docker images are built for the backend (Python/Flask), frontend (React Native), and AI service (TensorFlow).
3. Images stored in AWS ECR (Elastic Container Registry).
4. Deployment to staging for UAT (User Acceptance Testing).
5. **Manual approval** required for production deployment.
6. Production deployment via **Blue-Green strategy** to ensure zero downtime.

### 4.2 Deployment Strategies

Strategy	Description
Blue-Green	Traffic shifted to the new version after AI model and API stability checks.
Canary Release	Deploy to 5% of users first to validate real-time AI adjustments.

### 4.3 Rollback Strategy

- **Database Rollback:** Hourly backups stored in S3; restore via AWS Backup.
- **Versioned Containers:** Previous Docker images retained for 7 days.
- **Feature Flags:** Disable problematic AI recommendations instantly.
- **Monitoring Alerts:** Prometheus triggers rollback if API latency exceeds 2s.

## 5. Monitoring & Logging

### 5.1 Monitoring Tools

- **AI Performance:** Datadog for tracking recommendation accuracy and latency.

- **User Activity:** Custom dashboards in Grafana (e.g., "Time Efficiency %", "Active Users").
- **Error Tracking:** Sentry for real-time error alerts in mobile clients.

## 5.2 Logging Framework

- **Centralized Logs:** Logstash aggregates logs from mobile apps, AI models, and databases.
- **Retention:** 90 days for audit compliance.

# 6. Security Considerations

## 6.1 Authentication & Authorization

- **JWT Tokens** for user sessions.
- **Role-Based Access Control (RBAC):** Admins manage AI models; users access only their data.

## 6.2 Data Protection

- **Encryption:** User passwords will be hashed using **Argon2**. Sensitive data will be encrypted using **Elliptic Curve Cryptography (ECC)**.
- **Compliance:** GDPR adherence for EU users (data anonymization in AnalysisResults table).

## 6.3 DDoS Mitigation

- AWS Shield Advanced + Cloudflare for traffic filtering.
- Rate limiting on /api/recommendations endpoint.

# 7. Backup & Disaster Recovery Plan

## 7.1 Backup Strategy

- **Database:** Daily full backups + 15-minute incremental backups (MS SQL Server).
- **AI Models:** Versioned backups in S3 for training data and model weights.



## 7.2 Disaster Recovery Plan

- **Multi-Region Failover:** Deploy read replicas in EU (Frankfurt) and Asia (Singapore).
- **RTO:** <10 minutes (automated Kubernetes pod restarts).
- **RPO:** <5 minutes (latest incremental backup).

## 8. Risk Mitigation

Risk	Mitigation Strategy
High Traffic Overload	Auto-scaling (2–10 EC2 instances based on CPU usage).
Incorrect AI Recommendations	Daily retraining of models with user feedback data.
Data Breach	Quarterly penetration testing + anomaly detection.
API Downtime	99.9% SLA with AWS ELB health checks every 30s.

## 9. Conclusion

This deployment plan ensures the "Time in My Pocket" application delivers secure, real-time AI recommendations while maintaining scalability and reliability. By leveraging AWS infrastructure, robust CI/CD pipelines, and proactive monitoring, the system aligns with functional requirements (e.g., FR-03, NFR-02) and constraints (C-01, C-03). Regular audits and user feedback loops will further refine the deployment strategy post-launch.